

08-29-00

PTO/SB/05 (2/98)

Please type a plus sign (+) inside this box → ☐**UTILITY
PATENT APPLICATION
TRANSMITTAL**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.

TI-29602

First Named Inventor or Application Identifier

Srinath Hosur

Title

Receiver Algorithm for The Length 4 CFC

Express Mail Label No.

EL547747050US

On Page 1 of the specification, before line 1, insert -This application claims priority under 35 USC § 119(e)(1) of provisional application number 60/151,178 filed 08/27/1999.--

APPLICATION ELEMENTS

See MPEP Chapter 600 concerning utility patent application contents

ADDRESS TO:Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. <input checked="" type="checkbox"/> *Fee Transmittal Form (e.g., PTO/SB/17) (Submit an original, and a duplicate for fee processing)	6. <input type="checkbox"/> Microfiche Computer Program (Appendix)
2. <input checked="" type="checkbox"/> Specification (preferred arrangement set forth below) [Total Pages 27] - Descriptive title of the Invention - Cross References to Related Applications - Statement Regarding Fed sponsored R&D - Reference to Microfiche Appendix - Background of the Invention - Brief Summary of the Invention - Brief Description of the Drawings (if filed) - Detailed Description - Claim(s) - Abstract of the Disclosure	7. <input type="checkbox"/> Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary) a. <input type="checkbox"/> Computer Readable Copy b. <input type="checkbox"/> Paper Copy (identical to computer copy) c. <input type="checkbox"/> Statement verifying identical of above copies
3. <input checked="" type="checkbox"/> Drawing(s) (35 USC d113) [Total Sheets 11]	ACCOMPANYING APPLICATION PARTS 8. <input type="checkbox"/> Assignment Papers (cover sheet & Documents(s)) 9. <input type="checkbox"/> 37 CFR §3.73(b) Statement (when there is an assignee) <input type="checkbox"/> Power of Attorney 10. <input type="checkbox"/> English Translation Document (if applicable) 11. <input type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449 <input type="checkbox"/> Copies of IDS Citations 12. <input type="checkbox"/> Preliminary Amendment 13. <input checked="" type="checkbox"/> Return Receipt Postcard (MPEP 503) (Should be specifically itemized) 14. <input type="checkbox"/> *Small Entity Statement(s) <input type="checkbox"/> Statement filed in prior application Status still proper and desired (PTO/SB/09-12) 15. <input type="checkbox"/> Certified Copy of Priority Document(s) if foreign priority is claimed 16. <input type="checkbox"/> Other:
4. <input type="checkbox"/> Oath or Declaration [Total Pages <input]<br="" type="checkbox"/> a. <input type="checkbox"/> Newly Executed (original or copy) b. <input type="checkbox"/> Copy from a prior application (37 CFR §1.63(d)) (for continuation/divisional with Box 17 completed) [Note Box 5 below] i. <input type="checkbox"/> DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR §1.63(d)(2) and 1.33(b).	
5. <input type="checkbox"/> Incorporation By Reference (useable if Box 4b is checked) The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.	
17. If a CONTINUING APPLICATION , check appropriate box and supply the requisite information below and in a preliminary amendment: <input type="checkbox"/> Continuation <input type="checkbox"/> Divisional <input type="checkbox"/> Continuation-in-part (CIP) of prior application No: / . Prior application information: Examiner _____ Group / Art Unit: _____	

18. CORRESPONDENCE ADDRESS

Customer Number or Bar Code Label


23494

(Insert Customer No. or Attach bar code label here)



Correspondence address below

NAME			
ADDRESS			
CITY	STATE	ZIP CODE	
COUNTRY	TELEPHONE	FAX	
	(972) 917-4365	(972) 917-4418	

Name (Print/Type)	Carlton H. Hoel	Registration No. (Attorney/Agent)	29,934
Signature		Date	8/28/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

EXPRESS MAIL Mailing Label No. **EL547747050US**

Approved for use through 09/30/2000. OMB 0651-0032

DATE: **08/28/2000**

Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL

Patent fees are subject to annual revision on October 1.

These are the fees effective October 1, 1997

Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12.

Complete If Known

Application Number	
Filing Date	08/28/2000
First Named Inventor	Srinath Hosur
Examiner Name	
Group / Art Unit	
Attorney Docket No.	TI-29602

TOTAL AMOUNT OF PAYMENT **(\$)** **\$690****METHOD OF PAYMENT**

- 1.
- ☒
- The Commissioner is hereby authorized to charge to the following Deposit Account,

Deposit Account Number

20-0668

Deposit Account Name

Texas Instruments Incorporated

- ☒
- Charge any additional fee required or credit any overpayment
- ☐
- Charge all indicated fees and any additional fee required or credit any overpayment

- 2.
- ☐
- Payment Enclosed:**

☐ Check
 ☐ Money Order
 ☐ Other
FEE CALCULATION**1. BASIC FILING FEE**

Large Fee Code	Entity Fee (\$)	Small Fee Code	Entity Fee (\$)	Fee Description	Fee Paid
101	790	201	395	Utility filing fee	\$690
106	330	206	165	Design filing fee	
107	540	207	270	Plant filing fee	
108	790	208	395	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) **(\$)** **690****2. EXTRA CLAIM FEES**

	Extra Claims	Fee from below	Fee Paid
Total Claims	1	-20** = 0	x 18 = \$00
Independent Claims	1	-3** = 0	x 78 = \$00
Multiple Dependent			= \$00

**or number previously paid, if greater; For Reissue, see below

Large Fee Code	Entity Fee (\$)	Small Fee Code	Entity Fee (\$)	Fee Description
103	18	203	11	Claims in excess of 20
102	78	202	41	Independent Claims in excess of 3
104	270	204	135	Multiple dependent claims in excess of 3
109	82	209	41	**Reissue independent claims over original patent
110	22	210	11	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) **(\$)** **\$00****FEE CALCULATION (continued)****3. ADDITIONAL FEES**

Large Fee Code	Entity Fee (\$)	Small Fee Code	Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension of time within second month	
117	950	217	475	Extension of time within third month	
118	1,510	218	755	Extension of time within fourth month	
128	2,060	228	1,030	Extension of time within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,320	241	660	Petition to revive - unintentional	
142	1,320	242	660	Utility issue fee (or reissue)	
143	450	243	225	Design issue fee	
144	670	244	335	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt.	
581	40	581	40	Recording each patent assignment per property (time number of properties)	
146	790	246	395	Filing a submission after final rejection (37 CFR 1.129(a))	
149	790	249	395	For each additional invention to be examined (37 CFR 1.129(b))	

Other fee (specify) _____

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

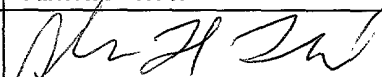
SUBTOTAL (3) _____

SUBMITTED BY

Typed or Printed Name

Carlton H. Hoel

Signature



Date

8/28/00

Complete (if applicable)

Reg. Number **29,934**

Deposit Account User ID

TIME DIVISION DUPLEX SYNCHRONIZATION

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from the following provisional applications: Serial Nos. 60/151,178, filed 08/27/99 and 60/189,350, filed 03/14/00. Copending application Serial No. 09/603,325, filed 06/26/00 discloses related subject matter. These applications have a common assignee.

BACKGROUND OF THE INVENTION

1. Field of the invention

The invention relates to communications, and more particularly to spread spectrum digital communications and related systems and methods.

2. Background

Spread spectrum wireless communications utilize a radio frequency bandwidth greater than the minimum bandwidth required for the transmitted data rate, but many users may simultaneously occupy the bandwidth. Each of the users has a pseudo-random code for "spreading" information to encode it and for "despreading" (by correlation) the spread spectrum signal for recovery of the corresponding information. Figure 2 shows a system block diagram, and Figures 3a-3b illustrates pseudo-random code plus a QPSK (quadrature phase-shift keying) encoder. This multiple access is typically called code division multiple access (CDMA). The pseudo-random code may be an orthogonal (Walsh) code, a pseudo-noise (PN) code, a Gold code, or combinations (modulo-2 additions) of such codes. After despreading the received signal at the correct time instant, the user recovers the corresponding information while the remaining interfering signals appear noise-like. For example, the interim standard IS-95 for such CDMA communications employs channels of 1.25 MHz bandwidth and a code pulse interval (chip) T_C of 0.8138 microsecond with a transmitted symbol (bit) lasting 64 chips. The recent wideband CDMA (WCDMA) proposal employs a 3.84 MHz bandwidth and the CDMA code length applied to each information symbol may vary from 4 chips to 256 chips. The

CDMA code for each user is typically produced as the modulo-2 addition of a Walsh code with a pseudo-random code (two pseudo-random codes for QPSK modulation) to improve the noise-like nature of the resulting signal. A cellular system as illustrated in Figure 4 could employ IS-95 or WCDMA for the air interface between the base station and the mobile user station.

A spread spectrum receiver synchronizes with the transmitter by code acquisition followed by code tracking. Code acquisition performs an initial search to bring the phase of the receiver's local code generator to within typically a half chip of the transmitter's, and code tracking maintains fine alignment of chip boundaries of the incoming and locally generated codes. Conventional code tracking utilizes a delay-lock loop (DLL) or a tau-dither loop (TDL), both of which are based on the well-known early-late gate principle.

In a multipath situation a RAKE receiver has individual demodulators (fingers) tracking separate paths and combines the results to improve signal-to-noise ratio (SNR), typically according to a method such as maximal ratio combining (MRC) in which the individual detected signals are synchronized and weighted according to their signal strengths. A RAKE receiver usually has a DLL or TDL code tracking loop for each finger together with control circuitry for assigning tracking units to received signal paths. Figure 5 illustrates a receiver with N fingers.

The UMTS (universal mobile telecommunications system) approach UTRA (UMTS terrestrial radio access) provides a spread spectrum cellular air interface with both FDD (frequency division duplex) and TDD (time division duplex) modes of operation. UTRA currently employs 10 ms duration frames partitioned into 15 time slots with each time slot consisting of 2560 chips. In FDD mode the base station and the mobile user transmit on different frequencies, whereas in TDD mode a time slot may be allocated to transmissions by either the base station (downlink) or a mobile user (uplink). In addition, TDD systems are differentiated from the FDD systems by the presence of interference cancellation at the receiver. The spreading gain for TDD

systems is small (8-16), and the absence of the long spreading code implies that the multi-user multipath interference does not look Gaussian and needs to be canceled at the receiver.

In currently proposed UTRA a mobile user performs an initial cell search when first turned on or entering a new cell; this search detects transmissions of base stations on the physical synchronization channel (PSCH) without any scrambling. The initial cell search by a mobile user must determine timing (time slot and frame) plus identify pertinent parameters of the found cell such as scrambling code(s).

For FDD mode the physical synchronization channel appears in each of the 16 time slots of a frame and occupies 256 chips out of the 2560 chips of the time slot. Thus a base station transmitting in the synchronization channel a repeated primary synchronization code of pseudo-noise of length 256 chips modulated by a length 16 comma-free code (CFC) allows a mobile user to synchronize by first synchronizing to the 256-chip pseudo-random code to set slot timing and then using the cyclic shift uniqueness of a CFC to set frame timing. Further, decoding the CFC by the mobile user reveals the scrambling code used by the base station.

In contrast, for TDD mode the physical synchronization channel only appears in one or two time slots per frame, so the length-16-CFC-modulated primary synchronization code does not easily apply. An alternative proposed TDD mode initial cell search employs a sum of a primary synchronization code (PSC) plus six secondary synchronization codes (SSCs); each code is a 256-chip pseudo-noise sequence, and the codes are orthogonal. In this proposal the initial cell search consists of slot synchronization, frame synchronization and code group identification with scrambling code determination. In particular, during slot synchronization the mobile user employs the PSC to acquire slot synchronization to the strongest cell (strongest received base station transmission): the PSC is common to all cells. A single matched filter (or any similar device matched to the PSC) may be used for detection.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

1. Overview

Preferred embodiment synchronization methods for UTRA-type TDD mode initial cell search use length 2 or 4 (or more) comma-free codes (CFCs) to encode both frame timing and base station code group information plus frame position for interleaved frames. The use of CFCs effectively removes the redundancy of straightforward coding of frame timing and code group in both time slots occupied by the physical synchronization channel. Some preferred embodiments use CFCs with alphabets generated from scaled sums of two or three QPSK-modulated secondary synchronization codes added to a primary synchronization code. Preferred embodiment spread spectrum communication systems incorporate preferred embodiment synchronization methods.

In preferred embodiment communications systems the base stations and the mobile users could each include one or more digital signal processors (DSP's) and/or other programmable devices with stored programs for performance of the signal processing of the preferred embodiment synchronization methods. The base stations and mobile users may also contain analog integrated circuits for amplification of inputs to or outputs from antennas and conversion between analog and digital; and these analog and processor circuits may be integrated on a single die.. The stored programs may, for example, be in ROM onboard the processor or in external flash EEPROM. The antennas may be parts of RAKE detectors with multiple fingers for each user's signals. The DSP core could be a TMS320C6x or TMS320C5x from Texas Instruments.

2. Preferred embodiments with 12 secondary synchronization codes

First preferred embodiment initial cell search methods by mobile users in a UTRA system in TDD mode employ the physical synchronization channel as illustrated in Figure 1a together with two-level frame interleaving. In particular, Figure 1a shows a 10 ms frame, 15 time slots per frame with 2560 chips per slot; the physical

Step 2. Frame synchronization and code-group identification plus frame position

During the second step the mobile user employs the secondary synchronization codes (SSCs) to find frame synchronization and identify one out of 32 code groups. Each code group is linked to a time offset of the synchronization channel within the slot, thus to a specific frame timing, and also is linked to a set of four scrambling codes (and basic midambles). To detect the position of the next synchronization slot, correlate PSC with the received signal at both 7 and 8 time slots after the slot detected in step 1. (The 7 and 8 arises because frames have 15 time slots.) The received signal at the positions of the synchronization slots is correlated with the PSC and all of the SSCs. These correlations may be done coherently over one or many time slots with phase correction provided by the correlation with PSC. The correlations recover the set of three b_i modulations for a time slot. Lookup in the table of Figure 1b yields the code group, time offset of the synchronization channel within the time slot, and the frame timing and interleave position.

Step 3. Scrambling code identification

During the third step the mobile user determines which of the four scrambling codes (and basic midambles) in the code group the cell is actually using. This may be by, for example, correlations with all four scrambling codes on the common control channel transmissions by the base station.

Preferred embodiment mobile users would have the PSC and SSCs (and scrambling codes and so forth) stored in memory and be programmed to execute the foregoing cell search.

3. Code minimum distance

The correlations in step 2 to find the QPSK modulation symbols $\{b_i\}$ are limited by signal-to-noise factors which relate to the minimum distance between possible codeword components being detected. The general minimum distance between two sums of three SSCs with QPSK modulation (e.g., $\sum b_i c_i / \sqrt{3}$ and $\sum b'_i c_i / \sqrt{3}$) occurs

when two out of the three modulation symbols are equal and when the differing symbols have one real and the other imaginary. Thus the minimum distance is $\sqrt{2}/\sqrt{3}$. Similarly, if the modulation symbols were constrained to be real (e.g., BPSK modulation), six SSCs would be required for the same modulation information, and the minimum distance would be $2/\sqrt{6} = \sqrt{2}/\sqrt{3}$, same as three SSCs with QPSK modulation. Contrarily, as Figure 1b illustrates, the preferred embodiments use only half of the QPSK modulation possibilities for each code set: either 0 or 2 imaginary modulations are used; never 1 or 3. This increases the minimum distance between CFC codeword components by $\sqrt{2}$ to $2/\sqrt{3}$. Further, Figure 1b also shows that decoding a single one of the four components of a CFC codeword determines the code group, frame timing, and frame interleave position because each component (set of b_1 , b_2 , and b_3) appears only once in the table.

However, the preferred embodiments using CFCs create gains over the simple use of sums of modulated SSCs in each time slot because part of the information is in the ordering (in time) of the sums of modulated SSCs (i.e., the ordering of the components of CFC codewords). Indeed, the use of six SSCs modulated by ± 1 in both time slots of a frame with one SSC modulation denoting which time slot and the other five modulated SSCs denoting the code group has redundancy in that the five modulated SSCs are repeated.

4. Preferred embodiments with 6 secondary synchronization codes

The second preferred embodiments apply for a physical synchronization channel with only one time slot per frame and with frame interleaving of depth 2. Figure 6 shows the codewords using two code sets of three SSCs to encode the 32 code groups. Basically, the first 16 length-4 codewords of Figure 1b are cut in half to form 32 length-2 codewords of Figure 6. This means that only two code sets are needed.

Of course, for two time slots per frame without frame interleaving the same codewords could be used.

Variable	Mean	SD	Min	Max
Age	34.5	10.2	21	55
Gender	Male			
Marital status	Married			
Education	High school			
Occupation	Teacher			
Income	1500	500	1000	2000
Health status	Good			
Smoking status	Non-smoker			
Alcohol consumption	No			
Stress level	Low			
Sleep quality	Good			
Exercise frequency	Low			
Dietary habits	Healthy			
Family size	3	1	2	5
Work hours	40	5	35	45
Commuting time	30	10	20	40
Childcare costs	500	100	400	600
Health insurance	Yes			
Life satisfaction	High			
Community involvement	Low			
Volunteer work	No			
Religious beliefs	Religious			
Political views	Conservative			
Environmental concerns	High			
Technology use	High			
Internet usage	High			
Smartphone ownership	Yes			
Social media use	High			
Online shopping	High			
Subscription services	High			
Home ownership	Yes			
Mortgage status	Paid off			
Property taxes	Low			
Neighborhood safety	High			
Local government	Effective			
Public services	Good			
Infrastructure	Good			
Public transportation	Good			
Parking availability	Good			
Local businesses	Good			
Community events	High			
Neighborhood diversity	High			
Local culture	Rich			
Historical landmarks	Many			
Local history	Rich			
Local economy	Strong			
Local industry	Diverse			
Local government	Effective			
Local politics	Active			
Local media	Active			
Local news	Active			
Local events	Active			
Local culture	Rich			
Local history	Rich			
Local economy	Strong			
Local industry	Diverse			
Local government	Effective			
Local politics	Active			
Local media	Active			
Local news	Active			
Local events	Active			
Local culture	Rich			
Local history	Rich			
Local economy	Strong			
Local industry	Diverse			
Local government	Effective			
Local politics	Active			
Local media	Active			
Local news	Active			
Local events	Active			
Local culture	Rich			
Local history	Rich			
Local economy	Strong			
Local industry	Diverse			
Local government	Effective			
Local politics	Active			
Local media	Active			
Local news	Active			
Local events	Active			
Local culture	Rich			
Local history	Rich			
Local economy	Strong			
Local industry	Diverse			
Local government	Effective			
Local politics	Active			
Local media	Active			
Local news	Active			
Local events	Active			
Local culture	Rich			
Local history	Rich			
Local economy	Strong			
Local industry	Diverse			
Local government	Effective			
Local politics	Active			
Local media	Active			
Local news	Active			
Local events	Active			
Local culture	Rich			
Local history	Rich			
Local economy	Strong			
Local industry	Diverse			
Local government	Effective			
Local politics	Active			
Local media	Active			
Local news	Active			
Local events	Active			
Local culture	Rich			</

The preferred embodiment for choosing code sets is as follows. All code sets are to be disjoint subsets of the set $\{C_0, \dots, C_{15}\}$. The number of code sets depends upon the number of groups to be represented. Each code set can give 8 length-4 comma free codewords. Thus when there are only 32 groups (and therefore 32 codewords), only 4 code sets are required, and choosing 4 disjoint subsets of $\{C_0, \dots, C_{15}\}$ is easy. For the embodiment where there are 256 codewords, 32 code sets are required. Not all code sets can be disjoint given that there are only 16 SSCs. In order to preserve the minimum distance, therefore allow a maximum of one SSC to be the same among any two code sets taken together. For example, $\{C_0, C_1, C_2\}$, $\{C_3, C_4, C_5\}$, AND $\{C_0, C_4, C_6\}$ are among the valid code sets which can be used for this case but $\{C_0, C_4, C_5\}$ is not.

The following Matlab program provides a simulation incorporating a preferred embodiment CFC decoding method for the code groups and code sets of Figures 7a-7b. The simulation inputs include signal-to-noise ratios, Doppler adjustments (external files), number of time slots of the synchronization channel used for decoding, and the number of simulation iterations. Each iteration randomly picks a code group for encoding and synthesizes signals and channel estimations. In particular, the matrix C for $q = 16$ corresponds to the code sets of Figure 7b with the subscript incremented by 1. And the 128 by 3 matrix C1 represents the 256 code

groups in pairs; the pairs differ by the sign of the second code. For example, the first two lines in Figure 7a have code groups labeled 0 and 1 with the difference being the sign of C_1 in each of the four time slots. More generally, if a code set consists of codes A, B, C; then the 8 code groups are, in terms of modulated sums as appear in the four time slots as in Figure 7a:

$$\begin{aligned} &\{A+B+C, A+B-C, -A-B+C, -A-B-C\}, \\ &\{A-B+C, A-B-C, -A+B+C, -A+B-C\}, \\ &\{jA+jB+C, jA+jB-C, -jA-jB+C, -jA-jB-C\}, \\ &\{jA-jB+C, jA-jB-C, -jA+jB+C, -jA+jB-C\}, \\ &\{jA+jC+B, jA+jC-B, -jA-jC+B, -jA-jC-B\}, \\ &\{jA-jC+B, jA-jC-B, -jA+jC+B, -jA+jC-B\}, \\ &\{jB+jC+A, jB+jC-A, -jB-jC+A, -jB-jC-A\}, \\ &\{jB-jC+A, jB-jC-A, -jB+jC+A, -jB+jC-A\}, \end{aligned}$$

Note each code group has the same internal sign change pattern from first to second to third to fourth elements. The corresponding four rows (three columns) of C_1 would be

A	B	C
A	B	C
A	C	B
B	C	A.

```
function pd_list = cfc_new(snr_list,num_slots,max_iter,fd,q,ng,foff)
```

```
if ng == 256
```

```
    if q == 30
```

```
        C = [1 2 3;4 5 6;7 8 9;10 11 12;13 14 15;16 17 18;19 20 21;
              22 23 24;25 26 27;28 29 30;1 4 7;10 13 16;19 22 25;2 5 8;
              11 14 17;20 23 26;3 6 9;12 15 18;21 24 27;1 5 9;7 11 15;
              13 17 21;16 20 24;19 23 27;22 26 30;3 5 7;6 8 10;9 11 13;
              12 14 16;18 20 22;21 23 25;24 26 28];
```

```
    elseif q==16
```

```
        C = [1 2 3; 4 5 6; 7 8 9; 10 11 12; 13 14 15; 1 4 7; 1 5 8; 1 6 9;
              1 10 13; 1 11 14; 1 12 15; 2 4 8; 2 5 7; 2 6 10; 2 9 11;
              2 12 13; 2 14 16; 3 4 9; 3 5 10; 3 6 7; 3 8 11; 3 12 14;
              3 13 16; 4 10 14; 4 11 13; 4 12 16; 5 9 12; 5 11 15; 6 8 12;
              6 11 16; 7 10 15; 8 10 16];
```

```
    else
```

```
        error('Only q=16 or q=30 currently');
```

```
    end
```

```

C1 = nchoosek(C(1,:),2);
C1 = [C1 flipud(C(1,:))];
C1 = [C1(1,:); C1];
for k=2:32
    Ck = nchoosek(C(k,:),2);
    Ck = [Ck flipud(C(k,:))];
    Ck = [Ck(1,:); Ck];
    C1 = [C1; Ck];
end
else
    error('Only 256 group case currently');
end

```

```

snr_list = 10.^(snr_list/10);

```

```

switch(fd)
case 0
    rayl_ch = ones(1,20);
case 5
    load /db/wcdma/200_Hz/fading_5_hz/old_sig_1.mat
    rayl_ch = old_sig_1.';
    clear old_sig_1;
    ch_len = length(rayl_ch);
case 20
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_20/rayl_sig
case 40
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_40/rayl_sig
case 80
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_80/rayl_sig
case 100
    %load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_100/rayl_sig
    %load

```

```

%/db/wcdma/32_KSPS/fading_1000_hz/NON_POW_CTRL/non_pow_ctrl_data_1_1
.mat
load /db/wcdma/200_Hz/fading_100_hz/old_sig_1.mat
%rayl_ch = old_sig_1(1:16:500000).';
rayl_ch = old_sig_1.';
clear old_sig_1;
ch_len = length(rayl_ch);
%load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_100/rayl_sig
case 460
    load /home/hosur/Matlab/CDMA/WCDMA/Nchmodel/Doppler_460/rayl_sig
case 1000
    rayl_ch = ones(1,20);

```



```

otherwise
    error('Unknown Doppler Frequency');
end

if (fd~=0)&(fd ~= 5)&(fd ~=100)
    ch_len = length(rayl_ch);
    rayl_ch = rayl_ch(:,1:20:ch_len); % subsample to get it at 16ksps
    ch_len = length(rayl_ch);
end

sc_rat = 0.5;
ssc_rat = sqrt(sc_rat);

pd_list=[];

tdec = [];
tpr = [ones(1,ceil(num_slots/4)); -ones(1,ceil(num_slots/4))];
tpr = reshape(tpr,1,prod(size(tpr)));
tpr = repeat(tpr,2);

eval(['load cr_corr_' int2str(q) '_' int2str(foff) 'khz.mat']);
pp = psc(1);
pscc = psc(2:q+1,:);
psc = psc.';

for snr = snr_list
    snr
    pd=0;
    ssc_rat_sc = sqrt(snr)*ssc_rat;
    for iter = 1:max_iter
        index = ceil(rand*ng);
        if index == 0
            index = 1;
        end

        cyc = ceil(rand*4);
        if cyc == 0
            cyc = 1;
        end

        %index = 1;
        %cyc=2;

        if (fd ~= 0)&(fd ~=1000)&(fd~=5)&(fd~=100)
            start = ceil(rand*(ch_len-8*num_slots));

```

```

if start == 0
    start = 1;
end
fading = rayl_ch(1,start:8:start+8*num_slots-1);
elseif (fd==5)|(fd==100)
    start = ceil(rand*(ch_len-num_slots));
    if start == 0
        start = 1;
    end
    fading = rayl_ch(1,start:start+num_slots-1);
elseif fd == 0
    fading = ones(1,num_slots);
elseif fd == 1000
    fading = (randn(1,num_slots)+j*randn(1,num_slots))/sqrt(2);
end

%C(index,:);
%size(psc)

k1 = ceil(index/2);
k2 = mod((index-1),2);
k3 = mod(k1-1,4);

tp = 1-2*bin_state(k2,2);

tv = ssc(:,C1(k1,:));
tva = tv(:,[1,2]).*tp(ones(q,1),:);
if k3
    tva = tva*j;
end
tva = sum(tva,2);
tvb = tv(:,3);

%disp('heh')
out_var = zeros(q,num_slots);
tv = zeros(q,4);

tpr1 = [tpr(cyc:length(tpr)) tpr(1:cyc-1)];
tpr1 = tpr1(1:num_slots);
for nk = 1:4
    ttem = tpr1(nk)*tva+((-1)^(nk+cyc))*tvb;
    out_var(:,nk:4:num_slots) = ttem(:,ones(1,num_slots/4));
end

tv = ssc rat sc*fading;

```

```

out_var = tv(ones(q,1),:).*out_var/sqrt(3);
out_var = out_var +...
    (1/sqrt(2))*(randn(q,num_slots)+j*randn(q,num_slots));

ch_est = zeros(1,num_slots);
tv = psc(C1(k1,:));
tva = tv([1,2]).*tp;
if k3
    tva = tva*j;
end
tva = sum(tva);
tvb = tv(3);
for nk = 1:4
    ch_est(1,nk:4:num_slots) = tpr1(nk)*tva+((-1)^(nk+cyc))*tvb;
end

ch_est = (sqrt(sc_rat*snr)*pp + ssc_rat_sc*ch_est/sqrt(3)).*fading...
    +sqrt(0.5)*(randn(1,num_slots)+j*randn(1,num_slots));
ch_est = ch_est./abs(ch_est);
%ch_est = fading;

% Now we receive the Comma Free Code and decide.

N = conj(ch_est(ones(q,1),:)).*out_var;

N1 = N(:,1:4:num_slots)+N(:,2:4:num_slots)-N(:,3:4:num_slots)-...
    N(:,4:4:num_slots);
N2 = N(:,1:4:num_slots)-N(:,2:4:num_slots)-N(:,3:4:num_slots)+...
    N(:,4:4:num_slots);
N3 = N(:,1:4:num_slots)-N(:,2:4:num_slots)+N(:,3:4:num_slots)-...
    N(:,4:4:num_slots);

N3 = real(N3);

N1 = sum(N1,2);
N2 = sum(N2,2);
N3 = sum(N3,2);

tv = zeros(4*ng,1);

tj = (mod([0:size(C1,1)-1]',4) > 0)*(-j);
tj(1:4:length(tj)) = 1;

%size(tj)
Nt1 = real(N1(C1(:,[1:2])).*tj(:,ones(1,2))));
Nt2 = real(N2(C1(:,[1:2])).*tj(:,ones(1,2))));

```

```
Nt3 = N3(C1(:,3));
```

```
tv(1:2:ng) = sum(Nt1,2)+Nt3;  
tv(2:2:ng) = -diff(Nt1,1,2)+Nt3;
```

```
tv(ng+1:2:2*ng) = sum(Nt2,2)-Nt3;  
tv(ng+2:2:2*ng) = -diff(Nt2,1,2)-Nt3;
```

```
tv(2*ng+1:2:3*ng) = -sum(Nt1,2)+Nt3;  
tv(2*ng+2:2:3*ng) = diff(Nt1,1,2)+Nt3;
```

```
tv(3*ng+1:2:4*ng) = -sum(Nt2,2)-Nt3;  
tv(3*ng+2:2:4*ng) = diff(Nt2,1,2)-Nt3;
```

```
dec_var = sum(tv,2);
```

```
[energy,decision] = max(dec_var);
```

```
cyc_h = floor((decision-1)/ng)+1;  
ind_h = decision - ng*(cyc_h-1);
```

```
%[ind_h index cyc_h cyc]
```

```
if (ind_h == index) & (cyc_h == cyc)  
    pd=pd+1;
```

```
end
```

```
end
```

```
pd_list = [pd_list pd/max_iter]
```

```
end
```

```
%keyboard
```

```
%eval(['save pd_list_cfcn_' int2str(fd) 'hz_coh pd_list snr_list']);
```

The preferred embodiment CFC decoding method appears in latter portion of the simulation. In particular, the matrix N is a 16 by number-of-time-slots array of correlations of the 16 synchronization codes with the simulation-generated received signals which incorporate a randomly selected code group (the "index" variable in the range of 1 to 256) and has a randomly selected cyclic shift (the "cyc" variable is the slot number in the range 1 to 4 (left to right in Figure 7a) of the first detected slot). Thus the decoding amounts to finding the values of "index" and "cyc". (Note

the subscript offset of 1: codes numbered 1-16 in the simulation correspond to codes C_0, \dots, C_{15} of Figures 7a-7b.)

Let $N(m,n)$ be the correlation of the m th synchronization code with the received (simulated) signal in the n th detected time slot of the synchronization channel. The preferred embodiment decoding defines $N1$, $N2$, and $N3$ as the column vectors of sums and differences of correlations over the detected time slots as follows:

$$\begin{aligned} N1(m) &= N(m,1) + N(m,2) - N(m,3) - N(m,4) \\ N2(m) &= N(m,1) - N(m,2) - N(m,3) + N(m,4) \\ N3(m) &= N(m,1) - N(m,2) + N(m,3) - N(m,4) \end{aligned}$$

Include any correlations for time slot numbers greater than 4 by treating $N(m,n)$ as $N(m, n \bmod 4)$. Thus $N1$, $N2$, $N3$ are just differing combinations of the correlations over the slots. Note that Figure 7a illustrates the significance of these $+$ $-$ patterns in the definitions of $N1$, $N2$, and $N3$; namely, for each code group slot k and slot $k+8$ of frame 1 (corresponding to successive slots in the simulation program) have signs for the first two codes opposite those of slots k and $k+8$ of frame 2 (which correspond to the next two successive slots in the simulation program). Thus $N1$ corresponds with frame alignment and $N2$ with a shift of one slot. Further, the third code has opposite signs in slots k and $k+8$ in both frames which $N3$ tracks. That is, if the received signal used a code group of just the q th code with signs $++--$, then $N1(q)$ will roughly be as follows: $+4$ if the first detected time slot is slot k of frame 1, 0 if the first detected time slot is slot $k+8$ of frame 1, -4 if the first detected time slot is slot k of frame 2, and 0 if the first detected time slot is slot $k+8$ of frame 2. Similarly, $N2(q)$ will roughly be 0 if the first detected time slot is slot k of frame 1, $+4$ if the first detected time slot is slot $k+8$ of frame 1, 0 if the first detected time slot is slot k of frame 2, and -4 if the first detected time slot is slot $k+8$ of frame 2.

Next, form the correlation combinations for the code groups of Figure 7a by defining $Nt1$, $Nt2$, and $Nt3$ as follows. $Nt1$ is a 128×2 matrix corresponding to the first two columns of the 128×3 code group matrix $C1$ but with an entry " m " (representing the m th synchronization code) in the k th row of $C1$ replaced by the real part of either $N1(m)$ or $-j \cdot N1(m)$, the former if $k-1$ is a multiple of 4, the latter otherwise. This $-j$

factor counters the j factor appearing in the first two columns of three quarters of the code groups as displayed in Figure 7a. Note that the t_j matrix in the simulation program represents this $-j$ factor incorporation. Similarly, $Nt2$ is the 128×2 matrix corresponding to first two columns of $C1$ but with an entry m replaced the real part of either $N2(m)$ or $-j \cdot N2(m)$. $Nt3$ is the 128×1 matrix corresponding to the third column of $C1$ with an entry m replaced by real part of $N3(m)$. As Figure 7a shows no j factors arise for $Nt3$ because the code groups do not use $\pm j$ with the third code. Thus the rows $Nt1$, $Nt2$, and $Nt3$ are the correlations

Then use $Nt1$, $Nt2$, and $Nt3$ to form a 1024×1 vector of the \pm combinations within a code group that appear in Figure 7a. In particular, a row in $C1$ corresponds to an adjacent pair of code groups, and each code group has four \pm combinations for the four time slots of Figure 7a. Thus there are $128 \cdot 8 = 1024$ combinations of correlations to assess. So define for m in the range 1 to 128 (corresponding to rows of $C1$ and pairs of code groups):

$$\begin{aligned} tv(2m-1) &= Nt1(m,1) + Nt1(m,2) + Nt3(m) \\ tv(2m) &= Nt1(m,1) - Nt1(m,2) + Nt3(m) \\ tv(256+2m-1) &= Nt2(m,1) + Nt2(m,2) - Nt3(m) \\ tv(256+2m) &= Nt2(m,1) - Nt2(m,2) - Nt3(m) \\ tv(512+2m-1) &= -Nt1(m,1) - Nt1(m,2) + Nt3(m) \\ tv(512+2m) &= -Nt1(m,1) + Nt1(m,2) + Nt3(m) \\ tv(768+2m-1) &= -Nt2(m,1) - Nt2(m,2) - Nt3(m) \\ tv(768+2m) &= -Nt2(m,1) + Nt2(m,2) - Nt3(m) \end{aligned}$$

The decoding then simply determines the maximum component value in the $tv()$ vector and then component number (the "decision" variable in the program) determines the cyclic shift as $cyc = \text{floor}((\text{decision}-1)/256) + 1$ and the code group index = decision - $256 \cdot (cyc-1)$. The "floor" function is the usual greatest-integer-not-greater-than function, and the code group index runs from 1 to 256 to correspond with the order in the lefthand ("code group") column of Figure 7a which labels the code groups in 8 intervals of numbers 0 to 31.

Of course, the preferred embodiment decoding method takes advantage of the structure of Figure 7a, and changing the code structure would lead to a

corresponding decoding method adjustment while retaining the formation of matrices of correlations with phase adjustments (\pm and j factors) combined to form all of the code group correlation sums followed by search for the maximum which yields the cyclic shift and code group indication.

In a more general encoding there could be q synchronization codes and each set of three codes (a code set) gives rise to 64 sums of QPSK-modulated codes (that is, each of the three codes in a code set can have one of 4 modulations: $+1$, -1 , $+j$, $-j$, so there are 4^3 combinations). Thus for length four CFCs with such sums as elements without repetition, there will be 16 possible code groups (4 sums in each code group), and further restriction to code groups with only an even number of $\pm j$ modulations in the sums halves the number of code groups to 8 per code set. Similarly, for length 2 CFCs there will be 32 code groups per code set, and further restriction to an even number of $\pm j$ modulations in the sums halves the number to 16 code groups. In fact, Figure 6 illustrates $q=6$, length 2 CFCs with two code sets and 32 code groups.

The CFC elements are detected in n time slots (if n is greater than the length of the CFC, then average over the detected time slots modulo the CFC length). Preferred embodiment decoding proceeds with steps such as:

(1) detect the correlations of each of the q synchronization codes with the received baseband signal in each of the n time slots; this yields $q*n$ correlations for n less than or equal to 2 or 4 (CFC lengths). For n greater than 2 or 4 average over time slots modulo 2 or 4 to have $q*2$ or $q*4$ correlations. Note that the array N in the foregoing simulation corresponds to these correlations prior to any averaging modulo 4. Denote the correlation with A of the received baseband signal in the first detected time slot (modulo 2 or 4) as $N(A,1)$, the correlation with C in the second detected time slot with C as $N(C,2)$, and so forth. For clarity, first consider length 4.

(2) form signed sums of these correlations. That is, define arrays $N1$, $N2$, and $N3$ as sums over the four time slots with two positive signs and two negative signs:

$$N1(A) = N(A,1) + N(A,2) - N(A,3) - N(A,4)$$

$$N_2(A) = N(A,1) - N(A,2) - N(A,3) + N(A,4)$$

$$N_3(A) = N(A,1) - N(A,2) + N(A,3) - N(A,4)$$

And similarly for other synchronization codes.

(3) Then define arrays Nt1, Nt2, and Nt3 which are row-indexed by the code groups used in the CFC and with entries corresponding to the N1, N2, and N3 with multipliers for the codes of the row's code group. For example, consider a code set of the three synchronization codes A, B, and C, which leads to 8 code groups including {A+B+C, A+B-C, -A-B+C, -A-B-C}, {jA+jC+B, jA+jC-B, -jA-jC+B, -jA-jC-B }, and so forth. (Such a set of 8 code groups corresponds to 8 adjacent lines in Figure 7a.) Thus define Nt1 and Nt2 as arrays with two columns (for the first and second codes in a code group) and Nt3 as an array with one column (for the third code in the code group):

$$\text{Nt1}(\{A+B+C, A+B-C, -A-B+C, -A-B-C\}, [1,2]) = [N1(A), N1(B)]$$

$$Nt2(\{A+B+C, A+B-C, -A-B+C, -A-B-C\}, [1,2]) = [N2(A), N2(B)]$$

$$Nt_3(\{A+B+C, A+B-C, -A-B+C, -A-B-C\}) = N_3(C)$$

and

$$\text{Nt1}(\{ |A+iC+B, |A+iC-B, -|A-jC+B, -|A-jC-B \}, [1,2]) = [-jN1(A), -jN1(C)]$$

$$\text{Nt2}(\{ |A+iC+B, |A+iC-B, -|A-jC+B, -|A-jC-B \}, [1,2]) = [-j\text{N2}(A), -j\text{N2}(C)]$$

$$Nt3(\{ |A+|C+B, |A+|C-B, -|A-|C+B, -|A-|C-B \}) = N3(B)$$

That is, for Nt1 and Nt2 the entry for a row is the pair of N1s and N2s corresponding to the two codes which maintain relative modulations within the code group; additionally, apply multipliers to make the first sum of the code group have positive real modulations for these two codes. The Nt3 entry is the N3 for the third code of the code group; this does not need a multiplier because the first sum of the code group always has a +1 modulation for the third code.

(4) for each code group form the 4 sums (which correspond to the first detected time slot falling into one of the four time slots of the length 4 CFC) of the components of Nt1 (or Nt2) plus Nt3. That is, code group {A+B+C, ...} leads to the four sums

$$N1(A) + N1(B) + N3(C)$$

$$N_2(A) + N_2(B) - N_3(C)$$

$$-N1(A) - N1(B) + N3(C)$$

-N2(A) - N2(B) - N3(C)

[illegible]

$$\begin{array}{l} -jN1(A) - jN1(C) + N3(B) \\ -jN2(A) - jN2(C) - N3(B) \\ jN1(A) + jN1(C) + N3(B) \\ jN2(A) + jN2(C) - N3(B) \end{array}$$

(5) find the maximum of the sums of step (4). The code group is that yielding the maximum, and the cyclic shift (the slot number of the first detected slot) is the number in the set of the four sums.

7. Preferred embodiments without frame interleaving

The fourth preferred embodiments apply to synchronization with non-interleaved frames but with 2 (or more) time slots per frame occupied by the physical synchronization channel. In particular, with C_1, C_2, \dots, C_6 being six SSCs, length 2 CFC codewords using QPSK modulation can be formed: $\langle (C_i + C_k)/\sqrt{2}, (C_i - C_k)/\sqrt{2} \rangle$, $\langle -(C_i + C_k)/\sqrt{2}, -(C_i - C_k)/\sqrt{2} \rangle$, $\langle j(C_i + C_k)/\sqrt{2}, j(C_i - C_k)/\sqrt{2} \rangle$, and $\langle -j(C_i + C_k)/\sqrt{2}, -j(C_i - C_k)/\sqrt{2} \rangle$ where C_i and C_k are a pair selected from the set C_1, C_2, \dots, C_6 without replacement. With six SSCs there are 15 pairs $\{C_i, C_k\}$ and thus 60 codewords of the foregoing type.

For the case of two time slots per (non-interleaved) frame and 32 code groups to identify with detection in a single time slot, using six SSCs with BPSK modulation and scaled by $\sqrt{6}$ as previously described, the minimum code distance between two possible detections is $2/\sqrt{6} = 0.816$. In contrast, with the foregoing length 2 CFC codewords the frame timing is inherent in the codeword component ($C_i + C_k$ is the first time slot and $C_i - C_k$ the second time slot) and the minimum code distance between two possible detections equals 1. Note that 30 of the 32 codewords needed can be with real (in phase) modulation and only two codewords need imaginary (quaternary) modulation. That is, 30 codewords may be of the form : $\langle (C_i + C_k)/\sqrt{2}, (C_i - C_k)/\sqrt{2} \rangle$ and $\langle -(C_i + C_k)/\sqrt{2}, -(C_i - C_k)/\sqrt{2} \rangle$, and only two imaginary modulated codewords need be used, for example, $\langle j(C_1 + C_2)/\sqrt{2}, j(C_1 - C_2)/\sqrt{2} \rangle$ and $\langle -j(C_1 + C_2)/\sqrt{2}, -j(C_1 - C_2)/\sqrt{2} \rangle$. And reflecting the greater minimum code distance, Figures 8-15 illustrate

the superior performance of such length 2 CFC codes versus the BPSK-modulated six SSCs.

Further, the computational complexity of the length 2 CFC is comparable to that of the six BPSK-modulated SSCs. In particular, for the six SSCs, 32 correlations are obtained performing length 8 correlations and the Fast Hadamard Transform is applied to these correlation values to obtain the correlations with the 6 SSCs and the PSC, requiring $8 \times 32 + 2 \times 16 \times \log_2 16 + 7 = 391$ complex additions. Again, the phase of the correlation with the PSC is used as a reference for the correlations with the 6 SSCs. This requires 28 real multiplications and 13 real additions. There are 64 possible combinations (codewords) of the 6 SSCs. Noting that some of these combinations are simply negatives of others and using other redundancies, the 64 combinations require 564 real additions. Averaging the 64 decision variables over K slots requires approximately 64 real additions per slot for large K. Selecting the maximum after averaging over K slots requires $(\log_2 64)/K$ compares per slot. This number is the same for both the CFC and the six SSC methods and for large K is very small. So neglect it in the analysis. Thus the six modulated SSC method requires 118 real additions per slot for computing, averaging the decision variables and selecting the maximum.

For the CFC, the multiplication by j is imply flipping the imaginary and real parts. Also, as noted above using $\pm \langle (C_i + C_k)/\sqrt{2}, (C_i - C_k)/\sqrt{2} \rangle$ produces 30 codewords so only $\pm \langle (C_1 + C_2)/\sqrt{2}, (C_1 - C_2)/\sqrt{2} \rangle$ are needed to complete the 32 required codewords. Again noting that some of these combinations are simply negatives of the others, the method requires 32 additions per slot to obtain the 64 decision variables. Averaging the 64 decision variables over K slots requires approximately 64 real additions per slot for large K. Again neglecting the compares required for selecting the maximum, which is the same for both methods, the length 2 CFC method requires 96 real additions per slot for computing, averaging the decision variables, and selecting the maximum.

8. Modifications

The preferred embodiments may be modified in various ways while retaining the features of comma-free codes (CFC) for synchronization in TDD systems. For example, the modulations b_i can be chosen from the set $\{e^{j\theta} e^{j\theta+j\pi/2} e^{j\theta+j\pi} e^{j\theta+j3\pi/2}\}$. In the preferred embodiment choosing b_i from the set $\{+1 +j -1 -j\}$ is equivalent to setting $\theta=0$.

Also, if the length-4 comma free code from the preferred embodiment is represented as $\{s_1 s_2 s_3 s_4\}$ where $s_k = \Sigma b_i c_i / \sqrt{N}$, any permutation of this code is also a comma free code, e.g. $\{s_2 s_4 s_3 s_1\}$, which is one such permutation, is also a comma free code. This implies that one can achieve the same comma free code performance by transmitting $\{s_2 s_4 s_3 s_1\}$ in sequence instead of $\{s_1 s_2 s_3 s_4\}$.

And for two length-4 comma free codes from the preferred set, say $\{s_1 s_2 s_3 s_4\}$ and $\{g_1 g_2 g_3 g_4\}$, then the codes formed by swapping any two elements between the two code words also form two length-4 CFC's, e.g. $\{s_1 g_2 s_3 s_4\}$ and $\{g_1 g_3 s_2 g_4\}$ are also valid comma free codewords which can be used instead of $\{s_1 s_2 s_3 s_4\}$ and $\{g_1 g_2 g_3 g_4\}$. Swapping the elements between three codewords forms a new set of three comma free codewords.

Further, using the construction method for CFCs made from two or three parallel SSCs, one can construct other CFC's with four or more SSC's in parallel.

Analogously, one can easily increase the length of the comma free codewords by concatenating two comma free codewords or portions of two comma free codewords. Similarly splitting the comma free codewords can also decrease the length.

Note that the use of comma-free codes for TDD permits all time slots (of the synchronization channel) contribute to the distance between codewords to avoid the problem of loss of diversity leading to a loss of codeword separation distance.

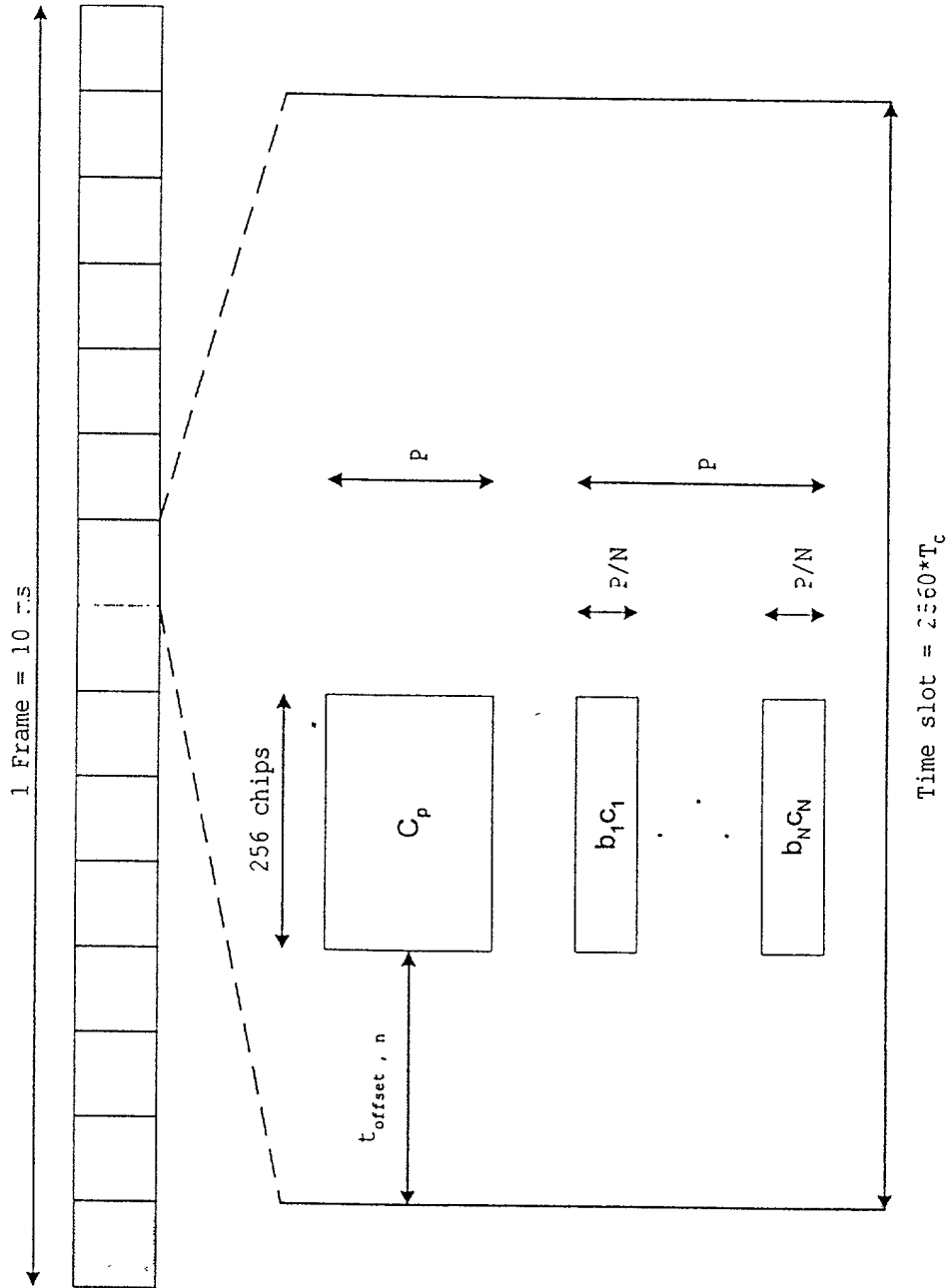
CLAIMS (29602)

What is claimed is:

1. A method for decoding received sums of QPSK-modulated spreading codes corresponding to elements of CFC codewords, comprising:
 - (a) despreading received sums of QPSK-modulated spreading codes with each of said spreading codes;
 - (b) forming linear combinations with coefficients ± 1 and $\pm j$ of the results of step (a), said combinations corresponding to possible sums as elements of CFC codewords;
 - (c) finding the maximum of said combinations of step (b);
 - (d) determining a codeword and cyclic shift from the results of step (c).

ABSTRACT OF THE DISCLOSURE

Decoding CFC codewords in the form of sums of QPSK-modulated spread symbols in a spread spectrum communications systems using comma-free codes with elements made of sums of QPSK-modulated secondary synchronization codes.



10

Figure 10: Scheme for Physical Synchronization channel PSC consisting of one primary sequence C_p and $N=3$ parallel secondary sequences in slot k and $k+8$

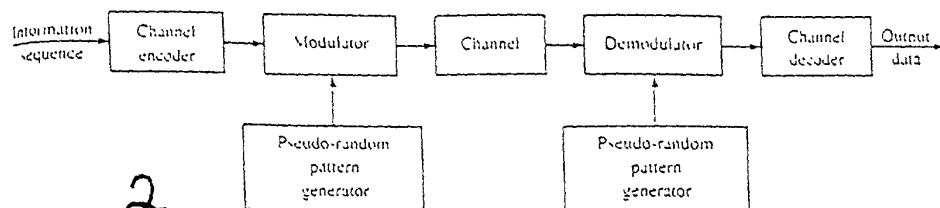
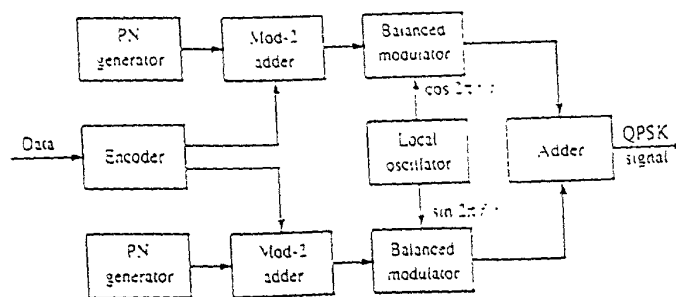
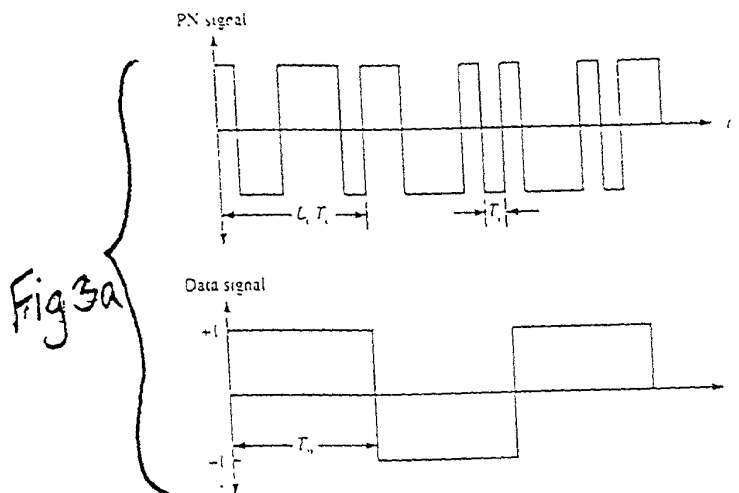


FIGURE 12-1-1 Model of spread spectrum digital communication system.



QPSK modulator

FIGURE 12-2 The PN and data signals (a) and the QPSK modulator (b) for a DS-SS spread spectrum system.

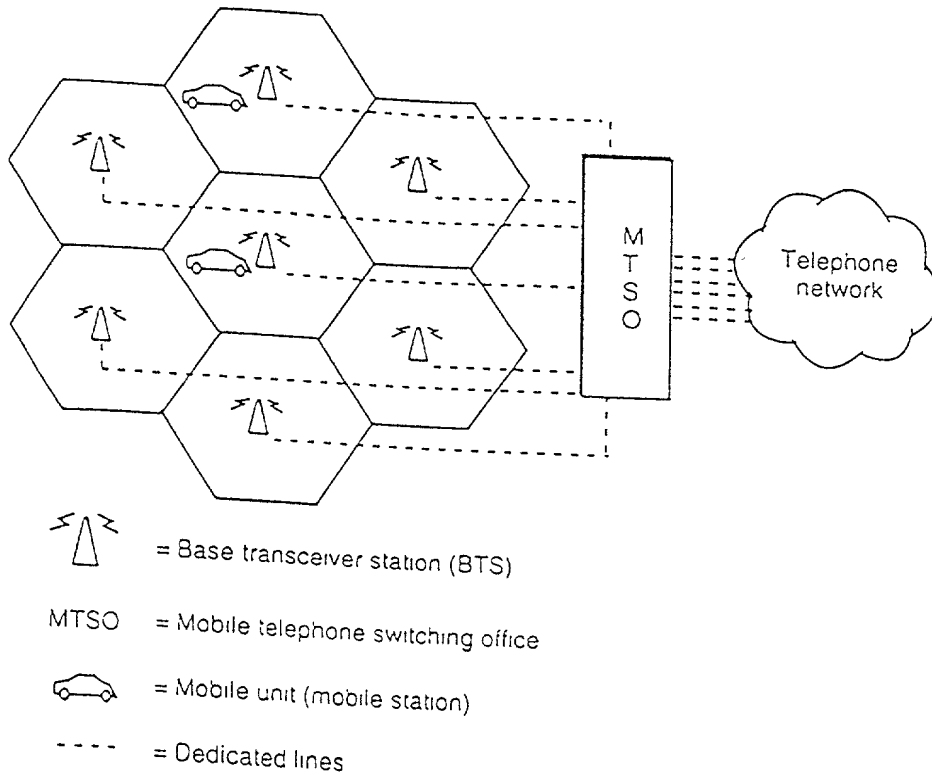


Figure 4-47
Cellular radio topology.

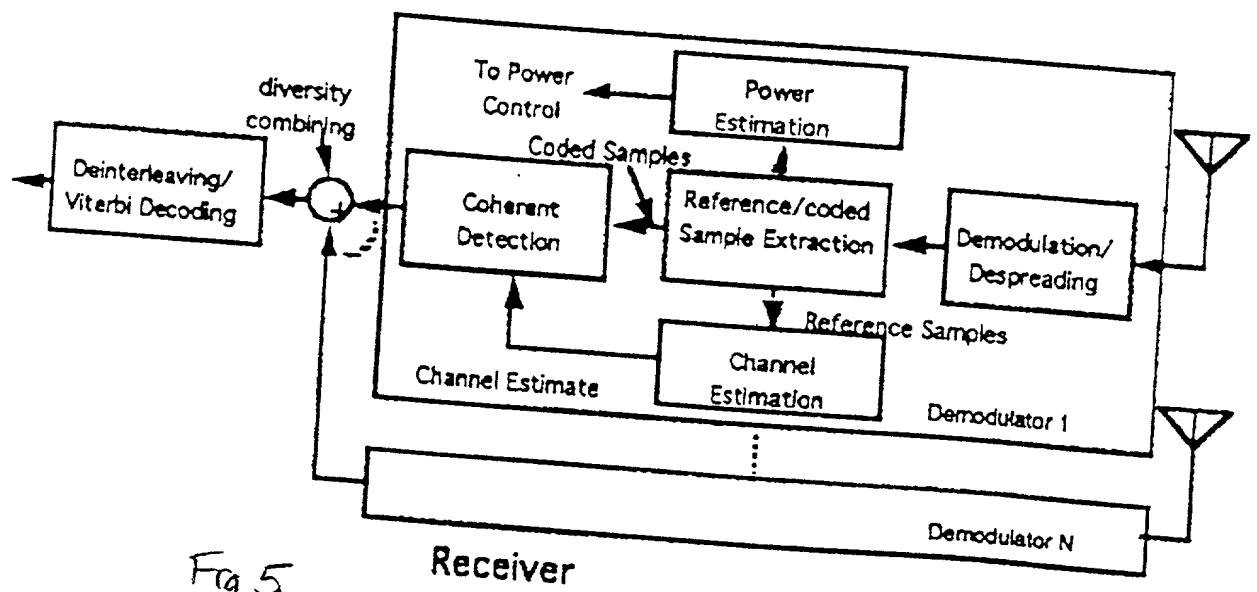


Fig 5

003230" 0624360

In addition to the information on code group three bits from SCH transport channel are transmitted to the UE with these codes.

Code Group	Code Set	Frame 1				Frame 2				Associated t_{offset}	Addl bits from SCH transport channel				
		Slot k		Slot k+8		Slot k		Slot k+8							
		C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_1			C_2	C_0	C_1	C_2
0	1	C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_1	C_2	t_0	000
1	1	C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_1	C_2	t_1	000
2	1	jC_0	jC_1	C_2	jC_0	jC_1	C_2	jC_0	jC_1	C_2	jC_0	jC_1	C_2	t_2	000
3	1	jC_0	jC_1	C_2	jC_0	jC_1	C_2	jC_0	jC_1	C_2	jC_0	jC_1	C_2	t_3	000
4	1	jC_0	jC_2	C_1	jC_0	jC_2	C_1	jC_0	jC_2	C_1	jC_0	jC_2	C_1	t_4	000
5	1	jC_0	jC_2	C_1	jC_0	jC_2	C_1	jC_0	jC_2	C_1	jC_0	jC_2	C_1	t_5	000
6	1	jC_1	jC_2	C_0	jC_1	jC_2	C_0	jC_1	jC_2	C_0	jC_1	jC_2	C_0	t_6	000
7	1	jC_1	jC_2	C_0	jC_1	jC_2	C_0	jC_1	jC_2	C_0	jC_1	jC_2	C_0	t_7	000
8	2	C_3	C_4	C_5	C_3	C_4	C_5	C_3	C_4	C_5	C_3	C_4	C_5	t_8	000
9	2	C_3	C_4	C_5	C_3	C_4	C_5	C_3	C_4	C_5	C_3	C_4	C_5	t_9	000
10	2	jC_3	jC_4	C_5	jC_3	jC_4	C_5	jC_3	jC_4	C_5	jC_3	jC_4	C_5	t_{10}	000
11	2	jC_3	jC_4	C_5	jC_3	jC_4	C_5	jC_3	jC_4	C_5	jC_3	jC_4	C_5	t_{11}	000
12	2	jC_3	jC_5	C_4	jC_3	jC_5	C_4	jC_3	jC_5	C_4	jC_3	jC_5	C_4	t_{12}	000
13	2	jC_3	jC_5	C_4	jC_3	jC_5	C_4	jC_3	jC_5	C_4	jC_3	jC_5	C_4	t_{13}	000
14	2	jC_4	jC_5	C_3	jC_4	jC_5	C_3	jC_4	jC_5	C_3	jC_4	jC_5	C_3	t_{14}	000
15	2	jC_4	jC_5	C_3	jC_4	jC_5	C_3	jC_4	jC_5	C_3	jC_4	jC_5	C_3	t_{15}	000
16	3	C_6	C_7	C_8	C_6	C_7	C_8	C_6	C_7	C_8	C_6	C_7	C_8	t_{16}	000
...
31	4	jC_{10}	jC_{11}	C_9	jC_{10}	jC_{11}	C_9	jC_{10}	jC_{11}	C_9	jC_{10}	jC_{11}	C_9	t_{31}	000
0	5	C_{12}	C_{13}	C_{14}	C_{12}	C_{13}	C_{14}	C_{12}	C_{13}	C_{14}	C_{12}	C_{13}	C_{14}	t_0	001
1	5	C_{12}	C_{13}	C_{14}	C_{12}	C_{13}	C_{14}	C_{12}	C_{13}	C_{14}	C_{12}	C_{13}	C_{14}	t_1	001
2	5	jC_{12}	jC_{13}	C_{14}	jC_{12}	jC_{13}	C_{14}	jC_{12}	jC_{13}	C_{14}	jC_{12}	jC_{13}	C_{14}	t_2	001
...
31	8	jC_5	jC_8	C_0	jC_5	jC_8	C_0	jC_5	jC_8	C_0	jC_5	jC_8	C_0	t_{31}	001
0	9	C_0	C_9	C_{12}	C_0	C_9	C_{12}	C_0	C_9	C_{12}	C_0	C_9	C_{12}	t_0	010
...
30	32	jC_9	jC_{15}	C_7	jC_9	jC_{15}	C_7	jC_9	jC_{15}	C_7	jC_9	jC_{15}	C_7	t_{30}	111
31	32	jC_9	jC_{15}	C_7	jC_9	jC_{15}	C_7	jC_9	jC_{15}	C_7	jC_9	jC_{15}	C_7	t_{31}	111

8x32 = 256 code groups

NOTE: The code construction using code sets 1 to 4 is exactly the same as for Case 2, and the additional bits from the SCH transport channel are "000". The code construction from code sets 5 to 32 is done in the same way with the additional bits for code sets 5 to 8 being "001", code sets 9 to 12 being "010", code sets 13 to 16 being "011", code sets 17 to 20 being "100", code sets 21 to 24 being "101", code sets 25 to 28 being "110", and code sets 29 to 32 being "111".

Fig. 7a

- Code set 1: C_0, C_1, C_2 .
- Code set 2: C_3, C_4, C_5
- Code set 3: C_6, C_7, C_8
- Code set 4: C_9, C_{10}, C_{11}
- Code set 5: C_{12}, C_{13}, C_{14}
- Code set 6: C_0, C_3, C_6
- Code set 7: C_0, C_4, C_7
- Code set 8: C_0, C_5, C_8 .
- Code set 9: C_0, C_9, C_{12} .
- Code set 10: C_0, C_{10}, C_{13}
- Code set 11: C_0, C_{11}, C_{14}
- Code set 12: C_1, C_3, C_7
- Code set 13: C_1, C_4, C_6
- Code set 14: C_1, C_5, C_9
- Code set 15: C_1, C_8, C_{10}
- Code set 16: C_1, C_{11}, C_{12} .
- Code set 17: C_1, C_{13}, C_{15}
- Code set 18: C_2, C_3, C_8
- Code set 19: C_2, C_4, C_9
- Code set 20: C_2, C_5, C_6
- Code set 21: C_2, C_7, C_{10}
- Code set 22: C_2, C_{11}, C_{13} .
- Code set 23: C_2, C_{12}, C_{15}
- Code set 24: C_3, C_9, C_{13}
- Code set 25: C_3, C_{10}, C_{12}
- Code set 26: C_3, C_{11}, C_{15}
- Code set 27: C_4, C_8, C_{11} .
- Code set 28: C_4, C_{10}, C_{14}
- Code set 29: C_5, C_7, C_{11} .
- Code set 30: C_5, C_{10}, C_{15}
- Code set 31: C_6, C_9, C_{14}
- Code set 32: C_7, C_9, C_{15}

Fig. 7b

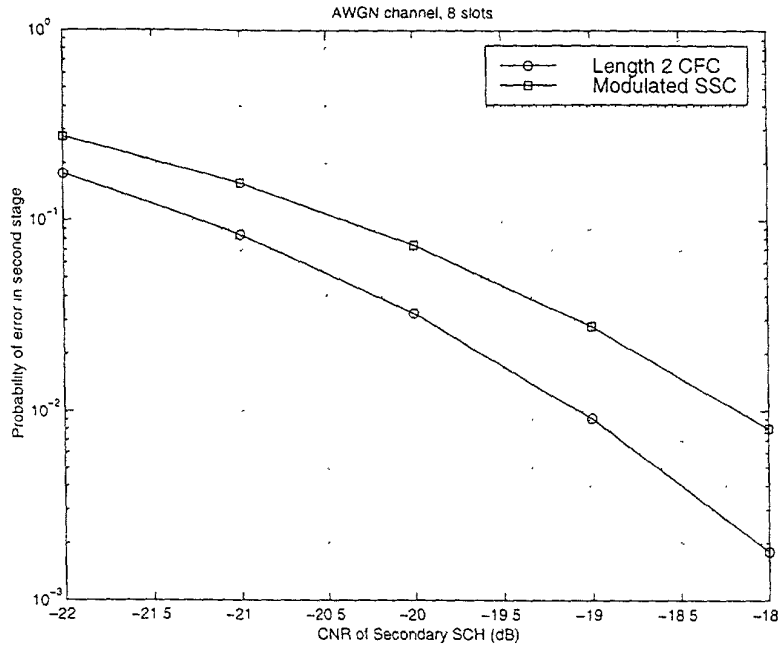


Figure 8. Figure comparing the Stage 2 performance of the Length 2 CFC to that of the Modulated SSC method for the AWGN case. The figure shows that the proposed method performs about 1.0dB better than the Modulated SSC method. 8 slots were used in Stage 2.

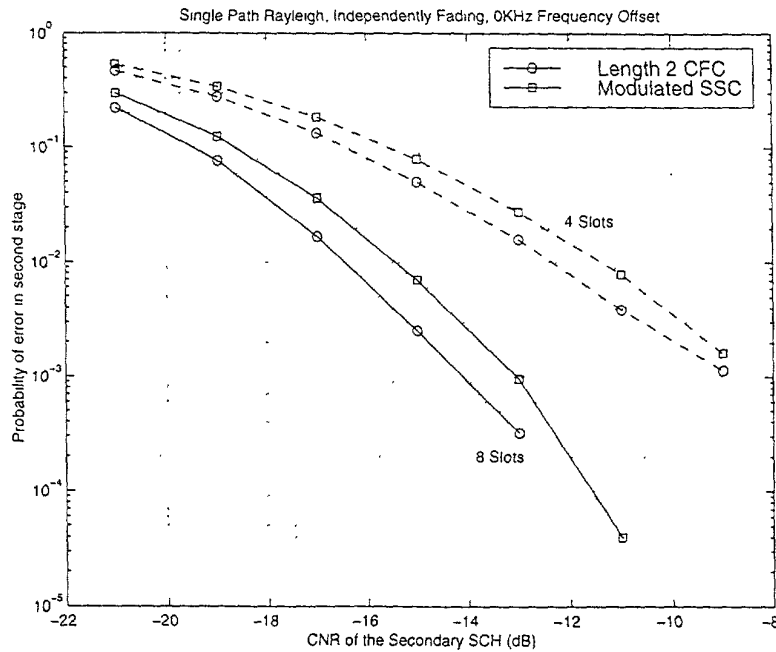


Figure 9. Figure comparing the Stage 2 performance of the length 2 CFC with that of the Modulated SSC scheme for the Rayleigh fading case. The figure shows that the performance of the length 2 CFC is better than that of the modulated SSC method by about 1.0dB for both 4 and 8 slots case. This is because the length 2 CFC has better distance than the modulated SSC method.

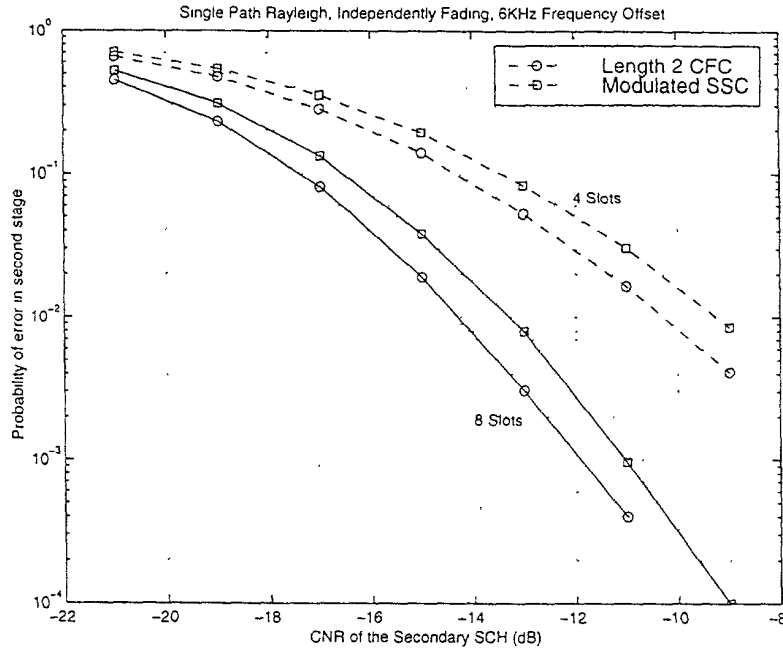


Figure 3. Figure comparing the Stage 2 performance of the length 2 CFC with that of the Modulated SSC scheme for the Rayleigh fading case under a 6KHz Frequency error. The figure shows that the performance of the length 2 CFC is still better than that of the modulated SSC method by about 1.0dB for both 4 and 8 slots case.

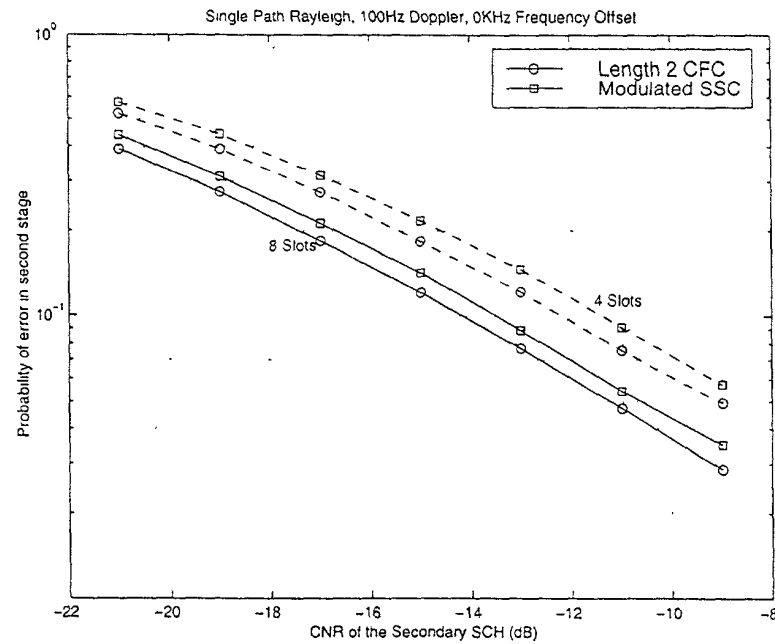


Figure 4. Figure comparing the Stage 2 performance of the length 2 CFC with that of the Modulated SSC scheme for the single path Rayleigh fading case, with a Doppler of 100Hz. There is no Frequency error. The figure shows that the performance of the length 2 CFC is still better than that of the modulated SSC method by about 1.0dB for both 4 and 8 slots case.

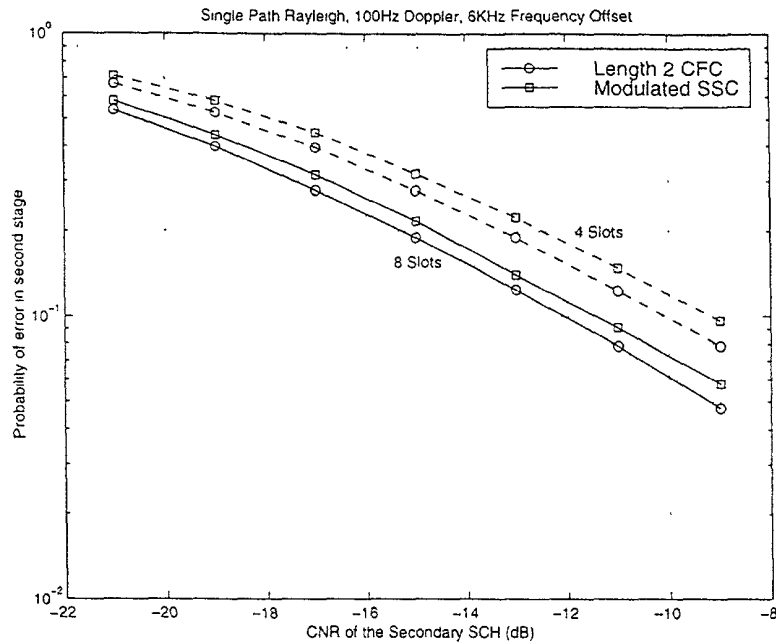


Figure 5. Figure comparing the Stage 2 performance of the length 2 CFC with that of the Modulated SSC scheme for the single path Rayleigh fading case, with a Doppler of 100Hz. The Frequency error is 6KHz. The figure shows that the performance of the length 2 CFC is still better than that of the modulated SSC method by about 1.0dB for both 4 and 8 slots case.

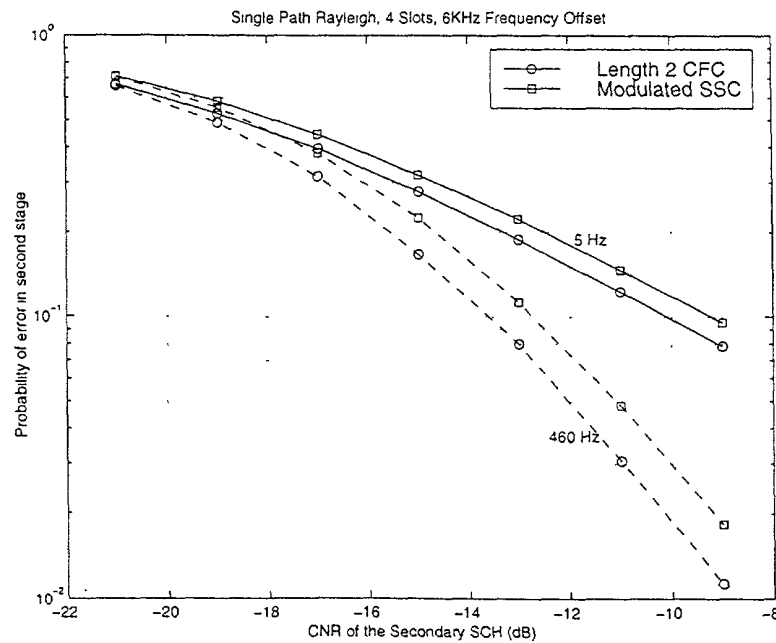
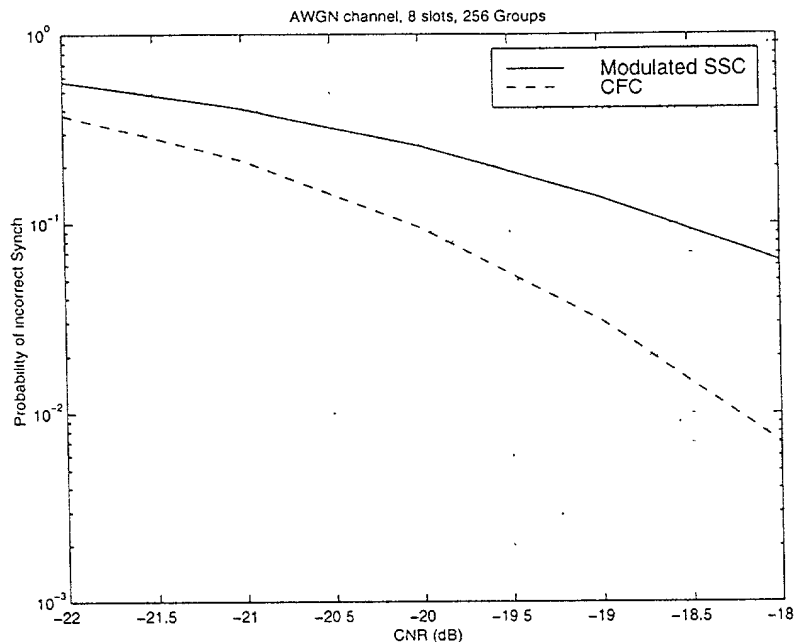
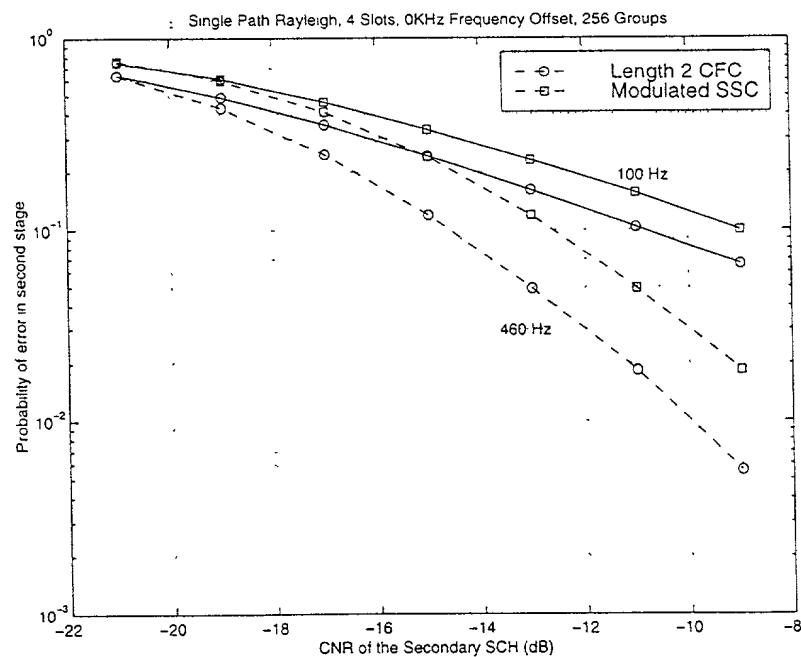


Figure 6. Figure comparing the Stage 2 performance of the length 2 CFC with that of the Modulated SSC scheme for the single path Rayleigh fading case, with Doppler's of 5Hz and 460Hz. The Frequency error is 6KHz and the number of slots was 4. The figure shows that the performance of the length 2 CFC is still better than that of the modulated SSC method by about



14
Figure 7. Figure comparing the Stage 2 performance of the length 2 CFC with that of the Modulated SSC scheme for the AWGN case. The number of long code groups is 256. The figure shows that the performance of the length 2 CFC is better than that of the modulated SSC method is greater than 1.5dB.



15
Figure 8. Figure comparing the Stage 2 performance of the length 2 CFC with that of the Modulated SSC scheme for the single path Rayleigh fading case, with Doppler's of 100Hz and 460Hz. There is no Frequency error and the number of slots was 4. The figure shows that the performance of the length 2 CFC is still better than that of the modulated SSC method by about 2.0dB.